

Reaktivní cache



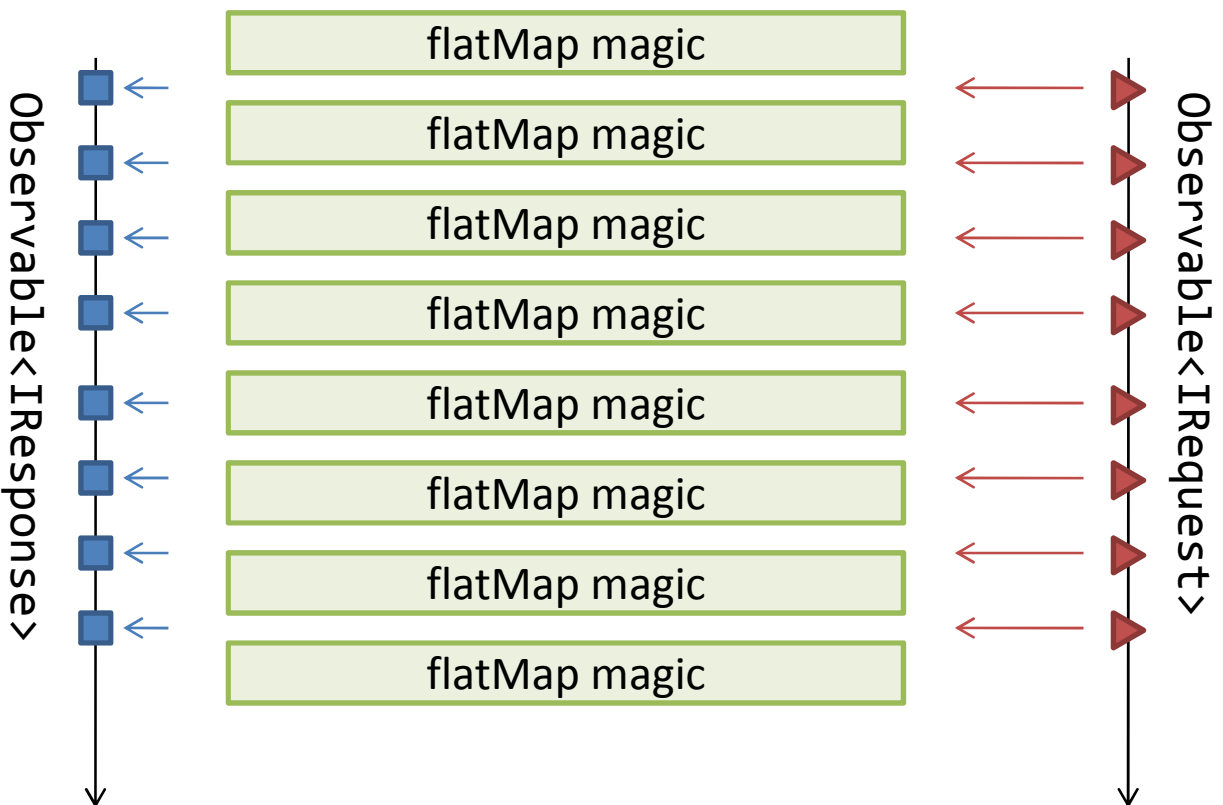
Vstup: IRequest



Od cache si slibujeme, že transformuje vstup (požadavek) na výstup.



Výstup: IResponse

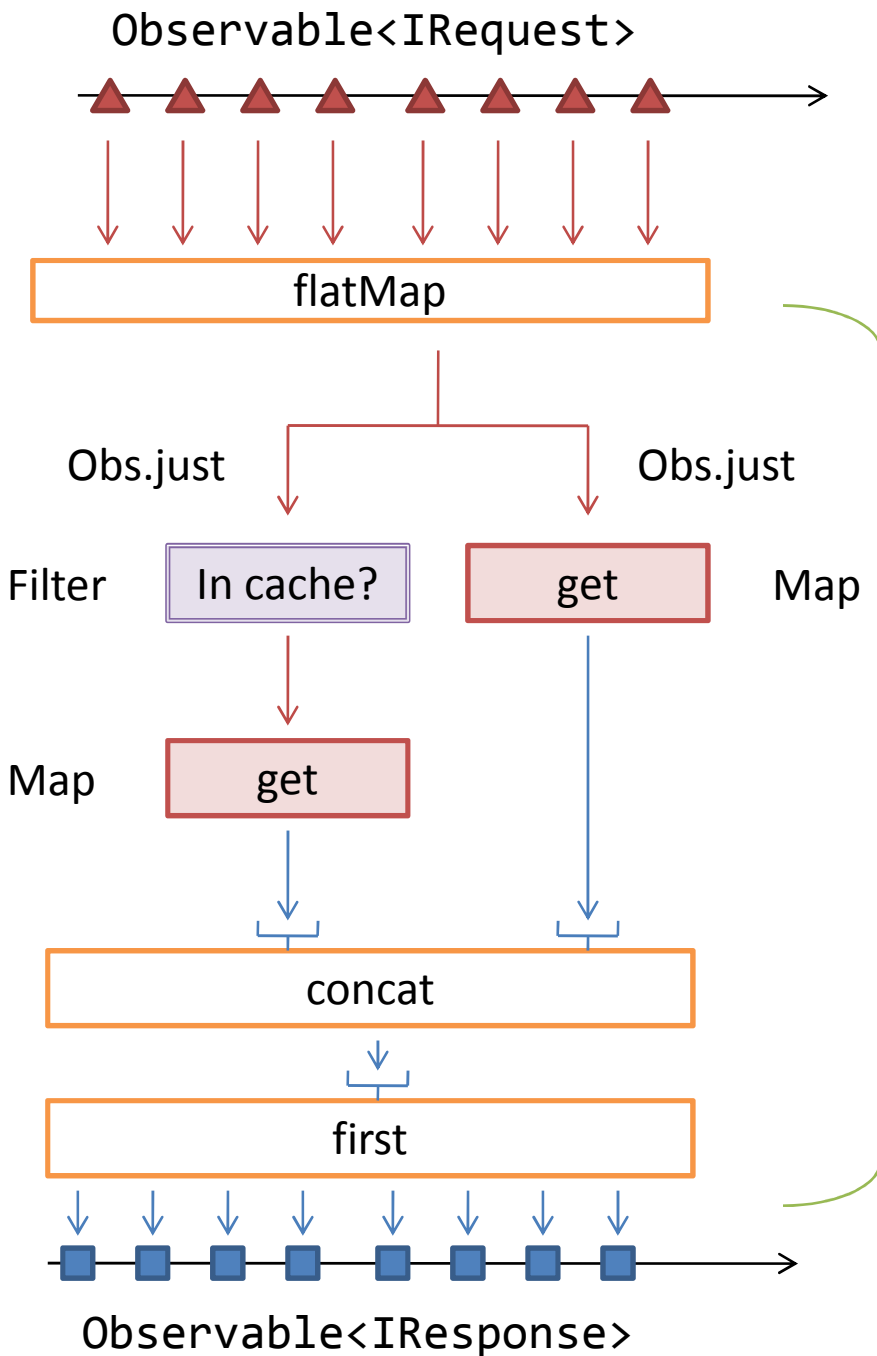


Vstup: IRequest

Následující slidy
vysvětlují z čeho
se „flatMap
magic“ skládá

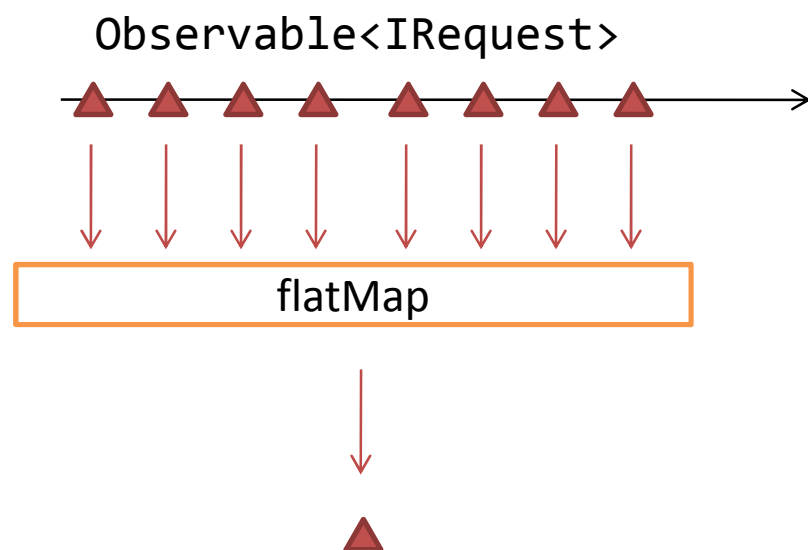


Výstup: IResponse



flatMap magic

Rozkreslená část diagramu reprezentující „flatMap magic“ se odehrává pro každý prvek. Kreslím případ pouze pro jeden z prvků, protože by se nevešli všechny najednou vedle sebe.



Chceme zkusit dvě větve:

1. První větev zkusí cache
2. Druhá větev načte soubor z disku

Z toho vyplývá, že chceme **2x nový observable**.

Observable<IRequest>



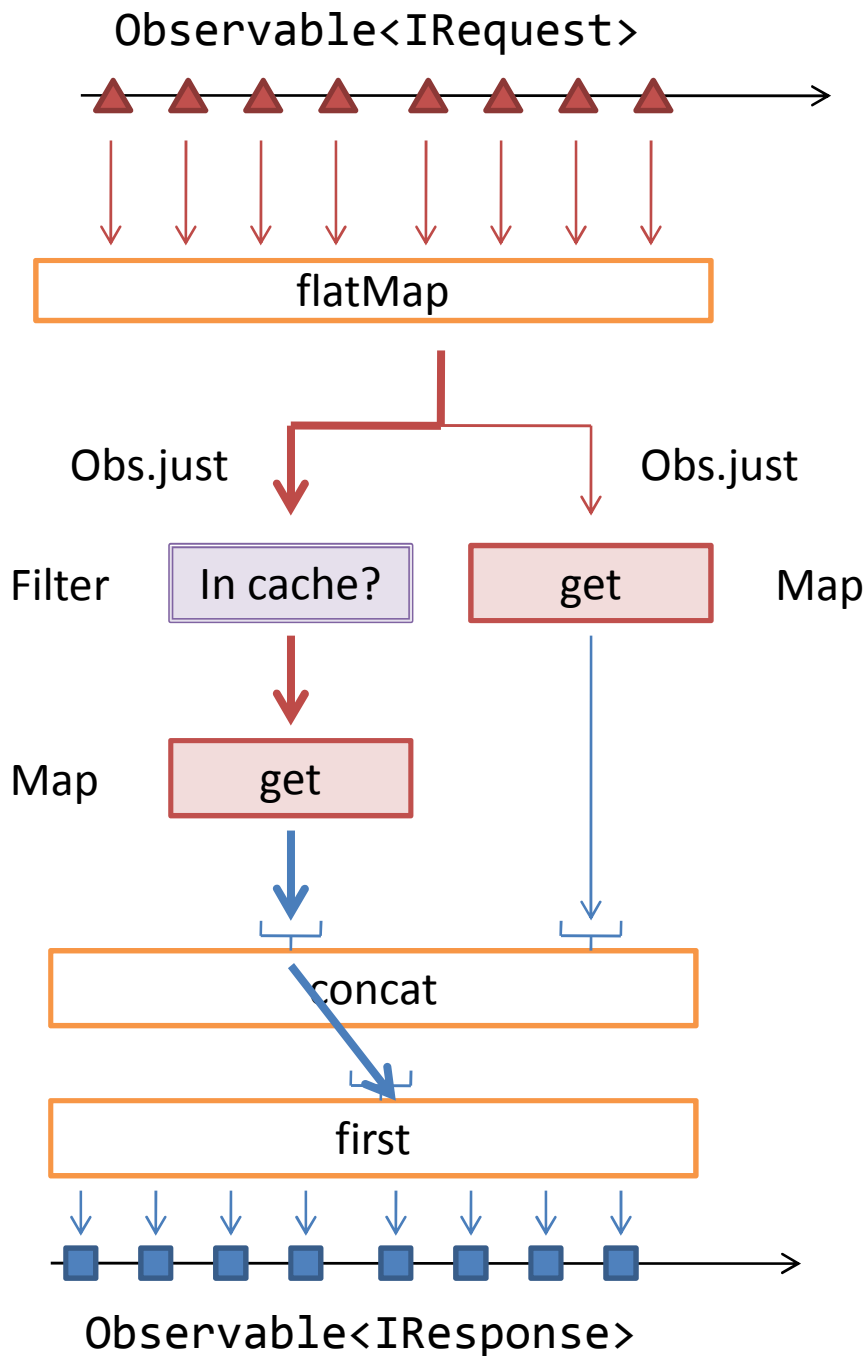
flatMap



Observable.just(▲)

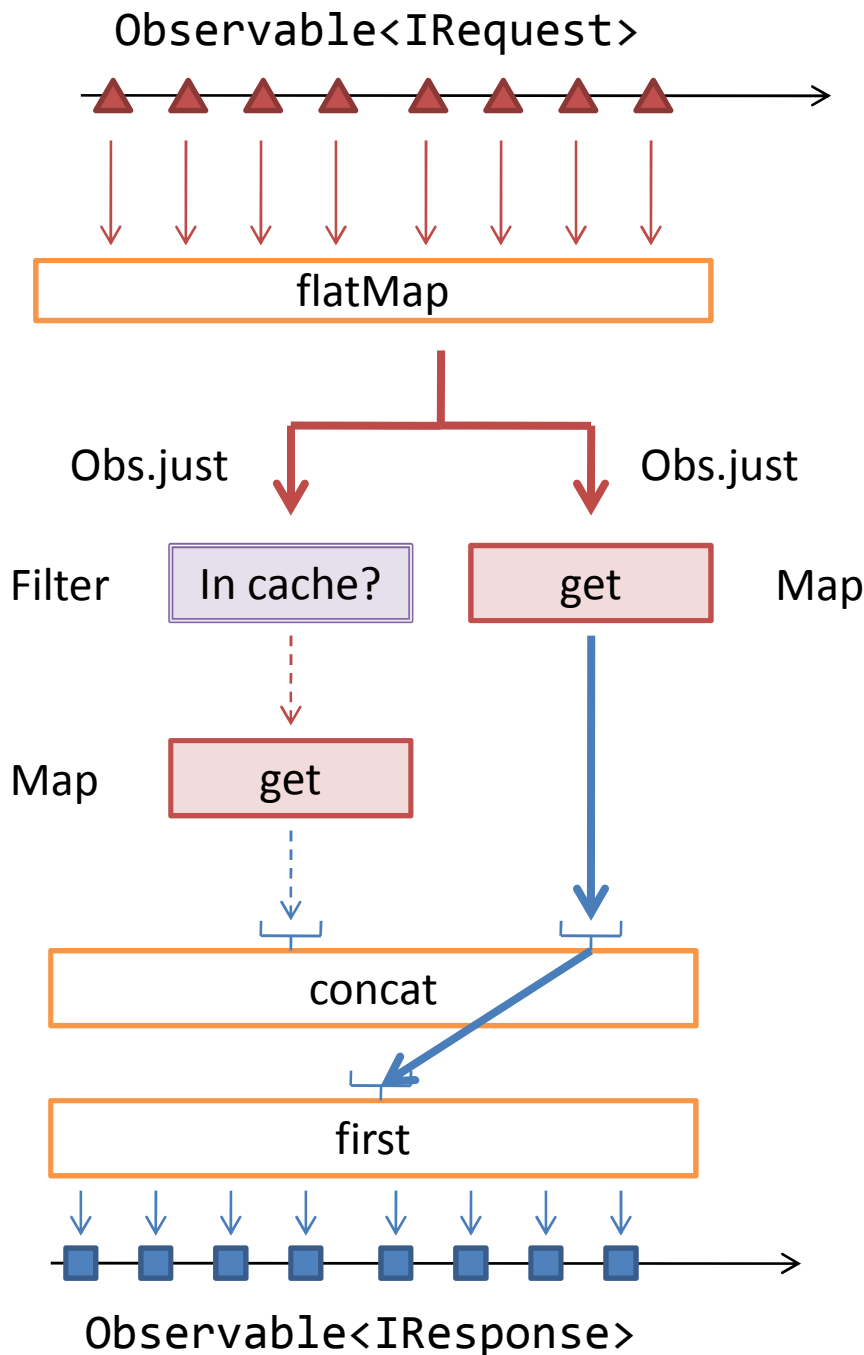
Observable.just(▲)





Případ 1. – request je v cache

1. Concat zavolá první ze dvou observable
2. Ten mu vrátí výsledek
3. First zajistí, že concat nebude pokračovat, protože již má prvek, který může operátoru first předat
4. First.toObservable vrací Observable, jehož obsah flatMap sloučí s ostatními do jednoho výstupního reaktivního streamu (výsledek tedy není `Observable<Observable<Iresponse>>`)



Případ 2. – request není v cache

1. Concat zavolá první ze dvou observable
2. Ten mu nevrátí výsledek
3. Concat zavolá druhý ze dvou observable
4. Ten vždy vrátí výsledek (pro neexistující soubor, vrátí např. `FileNotFoundException`)
5. `First.toObservable` vrací `Observable`, jehož obsah `flatMap` sloučí s ostatními do jednoho výstupního reaktivního streamu (výsledek tedy není `Observable<Observable<Iresponse>>`)